## Remarks

A. General Discussion of Differences Between Cited References and Present Invention

The presently claimed invention relates to software development of general stereoscopic haptic virtual environments, which can include creation or importation of a three-dimensional graphical virtual environments, generation of the left-eye, right-eye image pairs for the view plane as the graphical virtual environment is manipulated (e.g. rotating or touching objects within the scene graph), and interfacing the stereoscopic graphics virtual environment to shutter glasses for stereoscopic viewing and haptic interface devices for touching objects in such a way that the depth perception for stereoscopic viewing can be changed for individual users and the physical properties of the objects can be modified as needed, e.g. soft objects can be made hard and vice versa. Thus, by coupling force-feedback device with a stereoscopic display, the presently claimed invention makes it possible to create dynamic stereoscopic-haptic immersive virtual environment that allows seeing the three-dimensional virtual objects with their depth while touching them and feeling their physical properties.

Stewart[1] discloses a haptic interface method of modeling force feedback for a parametric free-form surface when the surface is touched (Col. 5, lines 14-16). The teachings of this reference are limited to determining when a free-form surface is touched (i.e. when the device collides with the surface (Col. 4, line 22 through Col. 6, line 36) and computing the force feedback (Col. 4, lines 12-21, and Col 6, line 37- Col 7, line 15) based on a friction coefficient and the distance of the tip of the haptic device from the surface within the elastic zone (i.e., "E-Zone" – Col. 5, lines 51-63). The remainder of the reference (Col 7, lines 16 to Col 9, line 10) simply provides computational examples of how the collision detection occurs for specifically claimed embodiments shown in Figures 10 and 11.

---

[1] U.S. Patent No. 5,694,013 issued to Stewart, *et al.*

There are a number of differences between the teachings of Stewart and the presently claimed invention:

1) "*stereoscopic haptic virtual environment*": Stewart teaches use of a head mounted virtual reality viewing device **18** to view a graphics environment consisting of a free-form parametric surface. (Col. 3, lines 33-36 - a surface S is displayed on a screen **16** (FIG. 2), or can be viewed in a virtual reality viewing device **18** that attaches directly to the head of the human operator **10** (as shown in **FIG. 1**)). Such a device is not a *stereoscopic* viewing device. Nowhere in Stewart is the term *stereoscopic* employed. Stewart's different mode of generating the parametric surface, as well as the reference's failure to address the issue of interfacing a stereoscopic viewing device (such as shutter glasses, to a graphics virtual environment), demonstrates that Stewart does not teach or suggest stereoscopic viewing. Effectively they are not generating the left eye – right eye scene pairs that are essential for stereoscopic viewing. The presently claimed invention generates left eye – right eye scene pairs and uses shutter glasses to achieve stereoscopic viewing.

With regard to claim 6, it is also not clear how to extend the disclosure in Isobe[2] of high-definition television system standards required for Isobe's "images" (Col. 4, lines 14-25) to convert these images to a general (standard independent) three-dimensional *stereoscopic graphical virtual environments* (scene graphs), even if a scene graph consisted simply of one parametric surface. Furthermore, the combination of Stewart and Isobe asserted as possible by the instant Action is not currently feasible and is rather impractical, *i.e.*, converting the *images* from the high-definition television system video signals *into a parametric surface* needed for the Stewart method is computationally too complex to make it practical. Implementation of such a combined scheme would require generalized three-dimensional segmentation algorithms, which at present do not exist. It has to be also noted that the objects made haptics only exist in a computer so that the use of any physical

[2] U.S. Patent No. 5,786,848 issued to Isobe, *et al.*

camera, such as the high-definition television camera taught by Isobe (Col. 4, lines 26-33), are of no use. In view of this, it is clear that converting these physical camera based *images* (based on high-definition television system standard) to general (standard independent) three-dimensional *stereoscopic haptic virtual environments* existing only in a computer (for which it is essential to generate the scene graphs) is not practical. In contrast, Applicants' specification describes a direct method for converting three-dimensional graphical virtual environments to three-dimensional *stereoscopic haptic* virtual environments.

2) "*stereoscopic graphics*": Stewart discloses a graphical virtual environment consisting of a computer generated (using CAD, CAM, or CAE tools) free-form parametric surface patch *only* (Abstract, Col. 2 lines 55-67, Col. 3 lines 25-29, Col. 4 lines 29-51, Col. 5 lines 14-15, FIG. 2, 4A, 4B, 4C, 5A, 5B, 9, 10, 11). The instant specification, in contrast, utilizes a scene graph comprised of a set of poly-mesh or parametric objects. This provides a general setting that permits working with *any* graphics scene consisting of a set of three-dimensional objects. Applicants' software allows *generation or importation* of any graphics virtual environment, transforming it into a *stereoscopic haptic* virtual environment that makes it possible to see the virtual objects with their depth while touching them. Furthermore, the present invention also generates stereoscopic-haptic virtual environment while performing basic graphics manipulations, such as rotating, translating, scaling the objects. This issue of manipulating graphical environment while touching virtual objects is not addressed in Stewart. It is clear from Stewart's computational examples (Col. 7, lines 16 to Col. 9, line 10, and FIGS. 10 and 11), that the free-form surface employed in their method is static when touched.

The presently claimed method includes generation of a small three-dimensional object (usually a sphere, but this can be any object such as a pen), a cursor or a proxy, to represent the tip of the haptic device (the "space pen", in Stewart terminology) in the virtual space. When the haptic device touches an object the cursor, a visual cue changes

color (and/or makes a sound). Without this cursor it is impossible for a user to know where they are in the virtual space, making it very difficult to actually work with the haptic device. Even though Stewart shows a pen on the surface in FIG. 2 and 4A, it is devoid of any teaching or suggestion of generating such a proxy to represent the space pen.

3) Haptic Process: Stewart's haptic process is based upon a force actuation check and force computation algorithms (Col. 4, lines 12-21). For the force actuation check there is a continuous loop to test if the haptic device collides with (i.e. touches) a virtual object, and if it does then the magnitude and direction of forces that are actuated for the physical haptic device are computed.

Stewart describes collision detection (or the force actuation check) algorithm for *only two specific cases*: **a)** when the parametric surface has infinite extension and thus divides the virtual space into two separate virtual spaces (Col. 1, lines 57-60, Col. 4, lines 38-49, FIG. 4B), in which case the algorithm is straightforward and simple, and **b)** when the parametric surface has limited extension (Col. 4, lines 50 - Col. 6, line 36, FIGS. 4C, 5A-B, 6, 7.) These algorithms are relatively simple and easy to implement, as can be seen from the patent (Col. 4, line 38 - Col. 6, line 36, FIG. 6 and 7). Applicants' work involves a general graphical (and stereoscopic) virtual environment, and thus requires generalized collision detection algorithms. These algorithms are more complex, as meeting real-time requirements for collision detection algorithms for general virtual environment is a challenging computational problem.

Stewart's force computation algorithm consists of computing the normal and the shear force at the point of collision between the space pen tip (i.e. the haptic device) and the surface (Col. 6, line 37 - Col. 7, line 15, FIG. 8). The normal force is computed based on the depth of device penetration (Col. 6, lines 51-53) *only* (they do not state varying of the stiffness coefficient as they do for the friction coefficient (Col. 7, lines 4-5)), while the present

invention also considers the stiffness and damping properties for computations. The sheer force in Stewart is also computed (Col. 7, lines 2-5) simply based on the static friction coefficient (that can be varied), while the present invention considers the velocity with which the device is moving (thus considering the dynamic friction as well). Thus, the present invention provides a solution that involves more complex and sophisticated force feedback computations.

4) Graphics to haptics synchronization: Stewart makes use of some synchronization terminology, but fails to provide details for synchronization implementation, however, one can infer that the synchronization is done by the controller interface 50 that provides communication between the processors 44, 46, 48 and the controller 28. Stewart does not mention the differences in the graphics and haptics rendering rates. This is important for synchronization of the graphics and haptic processes. Usually the haptic rendering is performed at a rate that is factor of tens faster than the graphics rendering. For example, in a haptics device suitable for use in the present invention, rendered may occur at up to 1 KHz, while the graphics can be rendered up to 60 Hz. These differences in the update rates and the complexity of the graphical virtual environment require a complex synchronization process for the haptic and graphic rendering algorithms to make sure that a stable haptic environment is created. None of these issues are addressed in Stewart.

5) Computational Set-up: The computational setup shown in Stewart's FIG. 3 (Col 3, line 48-Col. 4, line 11) implies that a CAD system and multiple processors are required for related computing. The presently claimed method can be executed on a PC, and does not require a CAD system or multiple processors. In contrast to the device controller *interface* (Stewart patent FIG. 3, item 28) to the computer (32), the present invention uses device drivers to send the haptic device tip (in Stewart terminology, the "space pen") position information (not joint positions information, item 30) to the computer and the computer sends back the forces to be applied to the haptic device tip (not joint forces, item 40).

B. Claim rejections under 35 U.S.C. §102

(a) Claims 2-5 were rejected as being anticipated by Stewart. Independent claim 2 recites:

*Method of creating a stereoscopic haptic virtual environment, comprising the steps of:*

*generating stereoscopic graphics and haptic scene components;*

*synchronizing the stereoscopic graphics and haptic scene components; and*

*presenting the synchronized stereoscopic graphics and haptic scene components to a user.*

(b) With regard to claim 2, the citations in the instant Action (at col. 2, lines 51-56, and FIGS. 3 and 4A) clearly indicate that Stewart's haptic interface method and graphical virtual environment consists of a *single* parametric surface that is generated using a CAD system. Stewart fails to disclose *generating* or viewing of *stereoscopic* components (e.g. left-eye, right-eye image pairs) as recited in claim 2. Please see Applicants' discussion points 1) and 2) above. Furthermore, Stewart's haptic interface method does not involve *generation of haptic scene components*; parametric information of the surface is used to compute the haptic interface (that is collision detection and generation of force). Col 1, lines 57-60, Col 4, line 38 to Col 6, line 36, FIG. 4A, B, C, 5A, B, 6, 7 provide additional support for this conclusion.

As described in Applicants' specification, the presently claimed invention creates a *three-dimensional graphical virtual environment* including any number of poly-mesh or parametric objects. Any graphical virtual environment may be converted into a stereoscopic graphical virtual environment. For example, the algorithm for the interlaced mode (that uses active shutter glasses to view the depth) involves moving the computational model of a camera left and right by customizable parallax amounts, and then using a stencil to draw the left and right corresponding virtual environment images followed by merging them to generate user specific depth perception for viewing the graphics virtual environment stereoscopically. In order to generate *haptic scene components*, all the parametric objects are converted into poly-mesh objects. These poly-mesh

objects are then transformed into a sharable scene graph that is used by both the graphics and haptic processes. In fact, the presently claimed method allows generation of stereoscopic graphics and haptics scene components even when rigid body dynamics (such as rotation, translation, scaling) are performed, as can be seen in Figures 6 (as update scene) and 7 (as update graphics, update haptics) of the instant specification.

With regard to the assertions made in paragraph 4 of the instant Action (which cites to col. 3, line 64 – Col. 4 line 1, FIG. 3, Col. 5, lines 25-31, and Col. 6, lines 42-46) related to *synchronization*, note 4) above discusses and demonstrates that the synchronization performed by Stewart's controller interface 50 is of a very *different* nature than the synchronization addressed in claims 2 and 5.

The synchronization process of the claimed invention accounts for differences in the graphics and haptics rendering rates. Usually the haptic rendering is done at a rate that is factor of tens faster than the graphics rendering. For example, a haptic device that might be employed in the present invention can be rendered at up to 1 KHz, while the graphics can be rendered up to 60 Hz. These differences in the update rates and the complexity of the graphical virtual environment require a complex synchronization process for the haptic and graphic rendering algorithms to make sure that a stable haptic environment is created. None of these issues are addressed in Stewart. (Please also refer back to the general rebuttal note section 1).) From a software development point of view, the presently claimed synchronization process is very complex, since the virtual environment is very complex, and as a result the stability of the haptic scene as well as the size of the graphics scene are addressed. Although not part of the present invention, Applicants have addressed some of these issues in the co-pending U.S. Patent Application Serial No. 09/844,635 entitled "Haptic Virtual Environments", assigned to the assignee of the present application.

Stewart's disclosure implies that the processor 1 (44) creates the surface (and the surface remains static through out the haptic interface process. This is confirmed by FIG. 6-11 and the fact that updating graphics is not addressed anywhere in the patent as well as the fact that there is no direct communication between processor 1 (44) and processor 2 (46)), the processor 2 (46) generates the points ($Q_i$) on the surface and computes distances (along with the signs) for the generated points, and the processor 3 (48) generates the space pen points ($P_i$) and computes forces to be actuated for the space pen. It is not clear how their haptic rendering loop works as no time parameters are provided. Usually the points ($P_i$) information for the haptic device positions are computed based on the timing information for the haptic rendering loop. For example, in the present invention these points are generated every 1ms (if we assume that the haptic servo-loop is 1KHz).

The references cited in paragraph 4 of the instant Action teach that the surface can be viewed on the monitor screen associated with the CAD system or the same image is seen using the virtual reality device (18). This by no means implies that the device 18 is a stereoscopic viewing device or that there exists a rendering algorithm for presenting a general stereoscopic graphics and haptic scene to a user. It appears that Stewart's method simply uses the CAD system's mono-display mechanism to present the surface that is generated by the CAD system, as it does not generate separate left and right eye images.

This is in contrast to the presently claimed method, wherein left and right eye images are rendered separately, thus the graphics is rendered twice to produce the stereoscopic effect. In addition to sending the computed force to the haptic device (thus making it possible for a user to feel the physical properties of the object being touched) the haptics scene component is visually presented using the cursor or a proxy (thus making it possible for a user to see the location at where the virtual object being touched). Stewart nowhere addresses these issues. Please see general rebuttal note 2) above for further discussion.

(c ) The instant Action also asserts that Stewart teaches all of the limitations of claims 3 and 4. Applicants respectfully disagree that the cited passages teach a special *visual cue* representing the tip of the haptic device, and even though Stewart shows a pen on the surface in FIG. 2 and 4A, it does not address the issue of generating such a proxy to represent the space pen anywhere in the patent. The visual component is simply the generated parametric surface that can be viewed on the monitor screen associated with the CAD system or on the virtual reality device (18).

General rebuttal note 3) above points out the differences between the collision detection algorithms of Stewart and the presently claimed invention and discusses the different methods by which the forces are computed. The different methods of computing forces using different parameters allow haptics to be presented in a different way to the user, for example, with the present method a user will feel dynamic friction and this will not be possible with Stewart's method. Thus, the presently claimed method achieves more complex sensations.

(d) With regard to claim 5, the portions of Stewart cited in the instant Action (col. 4, ll. 25-37) merely provides terminology that is used to describe the method of Stewart. Stewart is devoid of any detail regarding synchronization, however, one can infer (from Col. 3, line 48 - Col. 4, line 11, FIG. 3) that the synchronization is accomplished by the controller interface 50 that provides communication between the processors 44, 46, 48 and the controller 28. Stewart does not discuss the differences in the graphics and haptics rendering rates essential for synchronization of the graphics and haptic processes. Usually the haptic rendering is performed at a rate that is factor of tens faster than the graphics rendering. For example, a haptic device suitable for use in the present invention can be rendered at up to 1 KHz, while the graphics can be rendered up to 60 Hz. These differences in the update rates and the complexity of the graphical virtual environment require a complex synchronization process for the haptic and graphic rendering algorithms to make sure that a stable haptic environment is created. None of these issues are addressed in Stewart.

(e) In light of the foregoing, Applicants respectfully submit that Stewart fails to teach the limitations recited in claims 2-5, and respectfully requests reconsideration and withdrawal of this ground for rejection.

C. Rejection of claim 6 under 35 U.S.C. 103

Claim 6 was rejected as obvious of the combination of Stewart and Isobe. Applicants respectfully submit that the references are not properly combinable, and that improper hindsight has been used to make such a combination.

Extending the teachings of Stewart, which addresses a free-form parametric surface (or a surface patch), containing one three dimensional object, to cover the presently claimed method, which recites a *generalized scene graph* that includes a set of poly-mesh or parametric objects is not an obvious transformation.

It is not clear how to combine the teachings as suggested in the instant Action, i.e., interfacing high-definition television system standards as required for the *images* of Isobe (Col. 4, lines14-25) to a general (standard independent) three-dimensional parametric surface to make it stereoscopic virtual environment. In the general rebuttal note 1) above, the impracticality of such a combination is discussed. Even if it were possible to convert *a free-form parametric surface* utilized in Stewart into left eye - right eye image pairs to make it stereoscopic, Stewart's rendering algorithms will need further modification to make it stereoscopic and haptic at the same time. Further still, even this would be insufficient to make a general three-dimensional graphical virtual environment into a three-dimensional *stereoscopic haptic* virtual environment, as the collision detection algorithm of Stewart would be inoperable. The Stewart collision detection algorithm only works for a parametric surface and would not work for *a set of general three-dimensional poly-mesh objects*.

In light of at least the foregoing, Applicants respectfully submit that claims 2-6 are neither anticipated by nor made obvious by the cited art, and further examination and a notice to the allowability of said claims are earnestly solicited.

If questions remain, please call Applicants' attorney, collect, at the number given above. .

Respectfully submitted,

BHARTI TEMKIN, et al.,
Applicants

Date: January 24, 2005

By: John A. Hamilton

John A. Hamilton, Reg. No. 48,946
Attorney for Applicants

12001-104_RCE_012405